

One-Class SVM with Privileged Information and its Application to Malware Detection

Evgeny Burnaev

Skolkovo Institute of Science and Technology,
Building 3, Nobel st., Moscow 143026, Russia
Institute for Information Transmission Problems,
19 Bolshoy Karetny Lane, Moscow 127994, Russia,
Email: e.burnaev@skoltech.ru

Dmitry Smolyakov

Institute for Information Transmission Problems,
19 Bolshoy Karetny Lane, Moscow 127994, Russia,
Email: dmitry.smolyakov@iitp.ru

Abstract—A number of important applied problems in engineering, finance and medicine can be formulated as a problem of anomaly detection. A classical approach to the problem is to describe a normal state using a one-class support vector machine. Then to detect anomalies we quantify a distance from a new observation to the constructed description of the normal class. In this paper we present a new approach to the one-class classification. We formulate a new problem statement and a corresponding algorithm that allow taking into account a privileged information during the training phase. We evaluate performance of the proposed approach using a synthetic dataset, as well as the publicly available Microsoft Malware Classification Challenge dataset.

I. INTRODUCTION

Anomaly detection refers to the problem of finding patterns in data that do not conform to an expected behaviour. Anomaly detection finds extensive use in a wide variety of applications such as fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities [1], [2], [3]. A classical approach to anomaly detection is to describe expected (“normal”) behaviour using one-class classification techniques, i.e. to construct a description of a “normal” state using a number of examples, e.g. by describing a geometrical place of training patterns in a feature space. If a new test pattern does not belong to the “normal” class, then we consider it as an anomaly.

To construct a “normal” domain we can use well-known approaches such as the Support Vector Domain Description (SVDD) [4], [5] and the One-Class Support Vector Machine (One-Class SVM) [6], possibly combined with model selection for anomaly detection [7], resampling [8], ensembling of “weak” anomaly detectors [9] and extraction of important features using manifold learning methods [10], [11]. Both SVDD and One-Class SVM can be kernelized to describe a complex nonlinear “normal” class.

For the original two-class Support Vector Machine [12] Vapnik recently proposed a modification that allows taking into account a privileged information during the training phase to improve a classification accuracy [13]. Let us provide some examples of the privileged information. If we solve an image

classification problem, then as the privileged information we can use a textual image description. In case of a malware detection we can use a source code of a malware to get additional features for the classification. Such information is not available during the test phase (e.g. it could be computationally prohibitive or too costly to obtain), when we use the trained model for anomaly detection and classification, but can be used during the training phase.

In this work we combine these two concepts (SVMs and learning using privileged information) and propose an improved method to train One-Class SVM with privileged information, under the intuition that this additional information available at the training time can be utilized to better define the “normal” state. Through experiments on a synthetic data and some real data from the Malware Classification Challenge (see [14]), we show that the trained model with privileged information can perform better. However, we do not want to claim that the used setup of malware classification experiments indeed reflects a specificity of cyber security applications. Rather we demonstrate that the one-class classification with privileged information is useful for cyber security applications.

II. ONE-CLASS SVM AND SVDD

Below we briefly describe two classical approaches for one-class classification. We are given an i.i.d. sample patterns x_1, \dots, x_l from \mathbb{R}^n . The main idea of these algorithms is to separate a major part of sample patterns, considered to be “normal”, from those ones, considered to be “abnormal” in some sense.

A. One-Class SVM

In case of the original One-Class SVM [6] we consider those patterns to be abnormal, which are close to an origin of coordinates in a feature space.

Let us separate patterns using a hyperplane in a feature space defined by some feature map $\phi(\cdot)$ and a normal vector to the hyperplane w . We consider that a pattern x belongs to a “normal” class if $(w \cdot \phi(x)) > \rho$. In order to define the

hyperplane, i.e. the normal vector w and the value of ρ , we solve an optimization problem

$$\begin{aligned} & \frac{1}{2} \|w\|_{\ell_2}^2 + \frac{1}{\nu l} \sum_{i=1}^l \xi_i - \rho \rightarrow \min_{w, \xi, \rho} \\ & \text{s.t. } (w \cdot \phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0. \end{aligned} \quad (1)$$

Here ν is a regularization coefficient, ξ_i is a slack variable for the i -th pattern.

Optimization problem (1) is convex, therefore its solution coincides with that of the dual one:

$$\begin{aligned} & - \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j K(x_i, x_j) \rightarrow \max_{\alpha} \\ & \text{s.t. } \sum_{i=1}^l \alpha_i = 1, \quad 0 \leq \alpha_i \leq \frac{1}{\nu l}. \end{aligned} \quad (2)$$

Here the scalar product $(\phi(x_i) \cdot \phi(x_j))$ is replaced by the corresponding kernel function $K(x_i, x_j)$. Therefore as usual we do not need to know an explicit representation of $\phi(x)$ in order to solve the dual problem. Moreover, the solution of the primal problem can be represented through the solution of the dual problem, namely $w = \sum_{i=1}^l \alpha_i \phi(x_i)$.

Thanks to the structure of (2) if some $\alpha_i > 0$, then the pattern x_i belongs to the boundary of the “normal” domain [6], i.e. $(w \cdot \phi(x_i)) = \rho$. Thus the offset ρ can be recovered by exploiting the fact that for any $\alpha_i > 0$ the corresponding pattern x_i satisfies the equality

$$\rho = (w \cdot \phi(x_i)) = \sum_{j=1}^l \alpha_j K(x_j, x_i).$$

As a result the corresponding decision rule has the form

$$f(x) = \sum_{i=1}^l \alpha_i K(x_i, x) - \rho.$$

In case $f(x) > 0$ a pattern x is considered to belong to the “normal” class and vice versa. The absolute value $|f(x)|$ characterizes our confidence in this decision.

B. SVDD

Another approach to detect anomalies is to separate outlying patterns using a sphere [4], [5]. As before we denote by $\phi(\cdot)$ some feature map. Let a be some point in the image of the feature map and R be some positive value. We consider a pattern x to belong to a “normal” class, if it is located inside the sphere $\|a - \phi(x)\|_{\ell_2}^2 \leq R$. In order to find the center a and the radius R we solve the optimization problem

$$\begin{aligned} & R + \frac{1}{\nu l} \sum_{i=1}^l \xi_i \rightarrow \min_{R, a, \xi_i} \\ & \text{s.t. } \|\phi(x_i) - a\|_{\ell_2}^2 \leq R + \xi_i, \quad \xi_i \geq 0. \end{aligned} \quad (3)$$

Here ξ_i is a distance from the pattern x_i , located out of the sphere, to the surface of the sphere. On the face of it, the

variable R can be considered as a radius only if we require its positivity. However, it can be easily proved that this condition is automatically fulfilled if $\nu \in (0, 1)$ [5], and for $\nu \notin (0, 1)$ the solution of (3) is degenerate.

The dual problem has the form

$$\begin{aligned} & \sum_{i=1}^l \alpha_i K(x_i, x_i) - \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j K(x_i, x_j) \rightarrow \max_{\alpha} \\ & \text{s.t. } 0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad \sum_{i=1}^l \alpha_i = 1. \end{aligned}$$

As in the previous case we replace the scalar product $(\phi(x_i) \cdot \phi(x_j))$ with the corresponding kernel $K(x_i, x_j)$.

We can write out the solution of the primal problem using the solution of the dual problem

$$a = \sum_{i=1}^l \alpha_i \phi(x_i), \quad R = \|\phi(x_j)\|_{\ell_2}^2 - 2(a \cdot \phi(x_j)) + \|a\|_{\ell_2}^2,$$

where in order to calculate R we can use any x_j , such that $\alpha_j > 0$. Here $\|\phi(x)\|_{\ell_2}^2 = K(x, x)$, $(\phi(x) \cdot a) = \sum_{i=1}^l \alpha_i K(x_i, x)$ and $\|a\|_{\ell_2}^2 = \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j K(x_i, x_j)$.

The decision function has the form

$$f(x) = K(x, x) - 2 \sum_{i=1}^l \alpha_i K(x, x_i) + \|a\|_{\ell_2}^2 - R,$$

If $f(x) > 0$, then a pattern x is located outside the sphere and is considered to be anomalous. The absolute value $|f(x)|$ characterizes our confidence in this decision.

III. PRIVILEGED INFORMATION

Let us assume that on the training phase we have some privileged information: besides patterns $(x_1, \dots, x_l) \subset \mathbb{R}^n$ (original information) we also have additional patterns $(x_1^*, \dots, x_l^*) \subset \mathbb{R}^m$. This additional (privileged) information is not available on the test phase, i.e. we are going to train our decision rule on pairs of patterns $(x, x^*) \in \mathbb{R}^{n+m}$, but when making decisions we can use only test patterns $x \in \mathbb{R}^n$.

Let us discuss how this privileged information can be incorporated in the considered problem statement. In the original approaches to one-class classification we assume that the slack variables ξ_i , characterizing the distance from the patterns x_i to the separating boundary, are determined through the solution of the corresponding optimization problem (see (1) or (3)). Now let us assume that the slack variables can be modelled as $(\phi^*(x^*) \cdot w^*) + b^*$, where $\phi^*(\cdot)$ is a feature map in the space of privileged patterns. Thus, we assume that using the privileged patterns (x_1^*, \dots, x_l^*) we can refine the location of the separating boundary w.r.t. the sample of training objects.

A. One-Class SVM+

Let us modify problem statement (1) in order to incorporate the privileged information:

$$\begin{aligned} & \frac{\nu l}{2} \|w\|_{\ell_2}^2 + \frac{\gamma}{2} \|w^*\|_{\ell_2}^2 - \nu l \rho + \\ & \sum_{i=1}^l [(w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i] \rightarrow \min_{w, w^*, b^*, \rho, \zeta} \quad (4) \\ & \text{s.t. } (w \cdot \phi(x_i)) \geq \rho - (w^* \cdot \phi^*(x_i^*)) - b^*, \\ & (w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i \geq 0, \quad \zeta_i \geq 0. \end{aligned}$$

Here γ is a regularization parameter for the linear approximation of the slack variables, ζ_i are instrumental variables used to prevent those patterns, belonging to a “positive” half-plane, from being penalized. Note that if $\gamma \rightarrow \infty$, then the solution of (4) is close to the original solution of (1).

Let us write out a Lagrangian for (4):

$$\begin{aligned} L = & \frac{\nu l}{2} \|w\|_{\ell_2}^2 - \nu l \rho + \frac{\gamma}{2} \|w^*\|_{\ell_2}^2 \\ & + \sum_{i=1}^l [(w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i] - \sum_{i=1}^l \mu_i \zeta_i \\ & - \sum_{i=1}^l \alpha_i [(w \cdot \phi(x_i)) - \rho + (w^* \cdot \phi^*(x_i^*)) + b^*] \\ & - \sum_{i=1}^l \beta_i [(w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i]. \end{aligned}$$

Setting $\delta_i = 1 - \beta_i$, from the Karush – Kuhn – Tucker conditions we get that

$$\begin{aligned} w = & \frac{1}{\nu l} \sum_{i=1}^l \alpha_i \phi(x_i), \quad w^* = \frac{1}{\gamma} \sum_{i=1}^l (\alpha_i - \delta_i) \phi^*(x_i^*), \\ \delta_i = & \mu_i, \quad \sum_{i=1}^l \delta_i = \sum_{i=1}^l \alpha_i = \nu l, \quad 0 \leq \delta_i \leq 1. \end{aligned}$$

Using obtained equations, we now can formulate the dual problem

$$\begin{aligned} & - \frac{1}{2\nu l} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \\ & - \sum_{i,j} \frac{1}{2\gamma} (\alpha_i - \delta_i) K^*(x_i^*, x_j^*) (\alpha_j - \delta_j) \rightarrow \max_{\alpha, \delta} \\ & \text{s.t. } \sum_{i=1}^l \alpha_i = \nu l, \quad \sum_{i=1}^l \delta_i = \nu l, \quad 0 \leq \delta_i \leq 1, \quad \alpha_i \geq 0. \end{aligned}$$

Here we replace the scalar product $(\phi^*(x_i^*) \cdot \phi^*(x_j^*))$ with the kernel function $K^*(x_i^*, x_j^*)$. At the end, the decision function has the same form as in the case of the original One-Class SVM: $f(x) = \sum_{i=1}^l \alpha_i K(x_i, x) - \rho$.

B. SVDD+

Let us modify problem statement (3) in order to incorporate the privileged information:

$$\begin{aligned} & \nu l R + \frac{\gamma}{2} \|w^*\|_{\ell_2}^2 \\ & + \sum_{i=1}^l [(w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i] \rightarrow \min_{R, a, w^*, b, \zeta} \quad (5) \\ & \text{s.t. } \|\phi(x_i) - a\|_{\ell_2}^2 \leq R + [(w \cdot \phi^*(x_i^*)) + b^*], \\ & (w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i \geq 0, \quad \zeta_i \geq 0. \end{aligned}$$

If $\gamma \rightarrow \infty$, then the solution of (5) is close to the original solution of (3).

Let us write out a Lagrangian for (5):

$$\begin{aligned} L = & \nu l R + \frac{\gamma}{2} \|w^*\|_{\ell_2}^2 + \sum_{i=1}^l [(w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i] - \sum_{i=1}^l \mu_i \zeta_i \\ & + \sum_{i=1}^l \alpha_i [\|\phi(x_i) - a\|_{\ell_2}^2 - R - (w^* \cdot \phi^*(x_i^*)) - b^*] \\ & - \sum_{i=1}^l \beta_i [(w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i]. \end{aligned}$$

Setting $\delta_i = 1 - \beta_i$, from the Karush – Kuhn – Tucker conditions we get that

$$\begin{aligned} w^* = & \frac{1}{\gamma} \sum_{i=1}^l (\alpha_i - \delta_i) \phi^*(x_i^*), \quad a = \frac{1}{\nu l} \sum_{i=1}^l \alpha_i \phi(x_i), \\ \delta_i = & \mu_i, \quad \sum_{i=1}^l \alpha_i = \sum_{i=1}^l \delta_i = \nu l, \quad 0 \leq \delta_i \leq 1. \end{aligned}$$

Let us formulate the dual problem:

$$\begin{aligned} & \sum_{i=1}^l \alpha_i K(x_i, x_i) - \frac{1}{2\nu l} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \\ & - \sum_{i,j} \frac{1}{2\gamma} (\alpha_i - \delta_i) K^*(x_i^*, x_j^*) (\alpha_j - \delta_j) \rightarrow \max_{\alpha, \delta} \\ & \text{s.t. } \sum_{i=1}^l \alpha_i = \nu l, \quad \sum_{i=1}^l \delta_i = \nu l, \quad 0 \leq \delta_i \leq 1, \quad \alpha_i \geq 0. \end{aligned}$$

At the end, the decision function has the same form as in the case of the original SVDD: $f(x) = K(x, x) - 2 \sum_{i=1}^l \alpha_i K(x, x_i) + \|a\|_{\ell_2}^2 - R$.

IV. RELATED WORK

In principle some attempts to use a privileged information for one-class classification can be found in the literature. E.g. in [15] a feature space is divided in a number of subdomains and for each element from the training sample the number of the subdomain, to which this element belongs to, is used as a privileged information. In order to construct an anomaly detection rule the authors proposed to solve the following optimization problem:

$$\begin{aligned}
& \|w\|_{\ell_2}^2 + \frac{\gamma}{2} \|w^*\|_{\ell_2}^2 + \\
& \frac{1}{\nu l} \sum_{i=1}^l [(w^* \cdot \phi^*(x_i^*)) + b^*] - \rho \rightarrow \min_{w, w^*, b^*, \rho, \zeta} \quad (6) \\
& \text{s.t. } (w \cdot \phi(x_i)) \geq \rho - (w^* \cdot \phi^*(x_i^*)) - b^*, \\
& \quad (w^* \cdot \phi^*(x_i^*)) + b^* + \zeta_i \geq 0, \\
& \quad \zeta_i \geq 0.
\end{aligned}$$

One of the shortcomings of the problem statement (6) vs. the problem statement (4) is that the parameters ν and γ influence the regularization in a dependent manner, i.e. their contribution to the regularization can not be disentangled.

A similar framework is used in [16], where, as in the previous paper, the authors use a number of a subdomain as a privileged information. The main difference with paper [15] is that the SVDD algorithm underlies their approach. Finally the authors of [16] proposed to solve the following optimization problem in order to find the decision rule:

$$\begin{aligned}
& R^2 + \frac{\gamma}{2} (R^*)^2 \\
& + \frac{1}{\nu l} \sum_{i=1}^l [\|\phi^*(x_i^*) - a^*\|_{\ell_2}^2 - (R^*)^2] \rightarrow \min_{R, R^*, a, a^*, \zeta} \quad (7) \\
& \text{s.t. } \|\phi(x_i) - a\|_{\ell_2}^2 \leq R^2 + \|\phi^*(x_i^*) - a^*\|_{\ell_2}^2 - (R^*)^2, \\
& \quad \|\phi^*(x_i^*) - a^*\|_{\ell_2}^2 - (R^*)^2 + \zeta_i \geq 0, \\
& \quad \zeta_i \geq 0.
\end{aligned}$$

As in the previous example, the parameterization, used in (7), does not allow to control the regularization in the original feature space and in the space of privileged information independently. One more difference from the problem statement, proposed in the current paper, is another approach to model the slack variables ξ_i . In our approach we use the linear model, but in [16] the distance $\|\phi^*(x_i^*) - a^*\|_{\ell_2}^2 - (R^*)^2$ to the surface of the sphere in the privileged space is used for this purpose.

V. NUMERICAL EXPERIMENTS

In this section we describe performed numerical experiments. We provide results only for the original One-Class Support Vector Machine and its elaborated extension with privileged information, since results, based on SVDD, are comparable. This is not surprising, since when the Gaussian kernel is used the decision rules for the both methods are essentially the same, as it follows from the theoretical results of [5].

A. Data description

We perform experiments using both a two-dimensional synthetic data and some real data from the Microsoft Malware Classification Challenge (BIG 2015), see [14].

We generate the first synthetic dataset by the mixture of two-dimensional normal distributions, with unit covariance matrices and mean values $c_1 = (2, 2)$ and $c_2 = (-2, -2)$.

As a privileged information we use coordinates of a pattern after subtracting the nearest mean vector, i.e. $x^* = x - \arg \min_{c_1, c_2} (\|x - c_1\|, \|x - c_2\|)$.

We represent the second synthetic dataset by two circles of different radii with a common center. We generate each circle in a polar coordinate system, then Cartesian coordinates of a pattern are calculated. Values of an angle are generated from a uniform distribution $U[0, 2\pi)$. We generate values of a radius from a normal distribution $N(R, 0.5)$, where $R = 5$ for the external circle and $R = 0.5$ for the internal circle. We use Cartesian coordinates as features, and polar coordinates as a privileged information.

We generate the third synthetic dataset also in polar coordinates. Value of an angle is generated from the normal distribution $N(0, 1)$, value of a radius is generated from the normal distribution $N(0, 2\pi - \phi)$ for each point depending on the angle value ϕ in polar coordinates.

For some experiments we have to generate a sample with noise. In order to generate a noisy sample we

- generate a sample without noise,
- estimate bounds of the sample,
- generate noise using uniform distribution $U[a_1, a_2] \times U[b_1, b_2]$, where $a_1 = x_{1,\min} - 0.5 \cdot (x_{1,\max} - x_{1,\min})$, $a_2 = x_{1,\max} + 0.5 \cdot (x_{1,\max} - x_{1,\min})$, $b_1 = x_{2,\min} - 0.5 \cdot (x_{2,\max} - x_{2,\min})$, $b_2 = x_{2,\max} + 0.5 \cdot (x_{2,\max} - x_{2,\min})$.

Thus the region of the feature space, in which anomalies are located, includes all patterns of a “normal” data.

Examples of synthetic data are given in figure 1.

B. Proportion of discarded test sample patterns

In case of the original One-Class SVM a number of test patterns, marked by the decision rule as anomalous, depends on the regularization parameter ν [12], [4] in a very particular way: it tends to ν if the training sample size increases and both the test sample and the train sample are generated from the same distribution.

Using synthetic datasets let us check how the proportion of test sample patterns, marked as anomalous, depends on the regularization parameters ν and γ . We use the Gaussian kernel $K(x, x') = \exp(-\|x - x'\|^2 / \sigma^2)$ with $\sigma^2 = 2$ both for the original feature space, and for the privileged feature space.

In figure 2 we depict how the proportion of anomalous patterns depends on the parameters of the algorithm. We can notice that for small values of ν (significant regularization of the privileged space) this dependence is similar with that for the original One-Class SVM.

C. Accuracy of Anomaly Detection

For this experiment we generate samples with outliers using the approach, described in subsection V-A. The proportion of outliers is equal to 10%. For training we use an unlabeled sample. To assess the accuracy we calculate the area under the precision/recall curve using the test sample. Additionally we compare this accuracy with that of the original One-Class SVM. The main issue when performing comparison is how

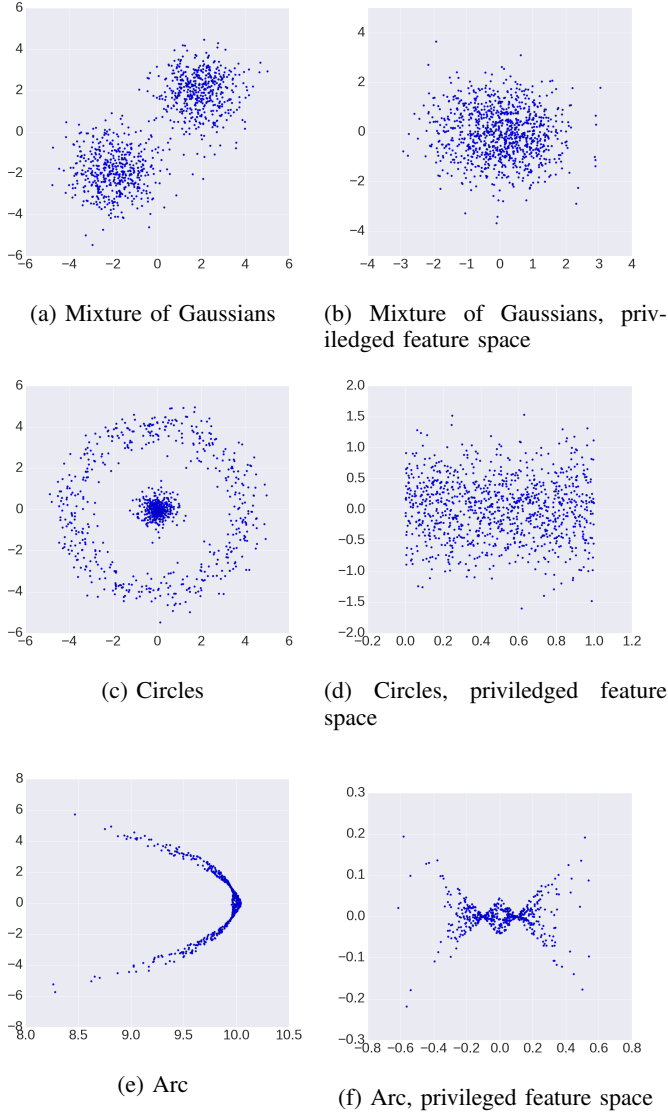


Fig. 1: Examples of synthetic datasets

to set values for the regularization parameters and the kernel widths. Let us comment on this issue:

- In order to tune the regularization parameter ν and the kernel width σ for the original One-Class SVM, we perform a grid search in order to maximize the area under the precision/recall curve, estimated by the cross-validation procedure. We denote obtained “optimal” values by ν_{opt} and σ_{opt} and provide the corresponding anomaly detection accuracy in figure 3.
- For the One-Class SVM+ we tune only the regularization parameter γ and the kernel width σ^* , responsible for the “privileged” part of optimization problem (4). As above, we optimize the area under the precision/recall curve, estimated by the cross-validation procedure. In this case we set values of the parameters ν and σ to ν_{opt} and σ_{opt} correspondingly, which are optimal for the original One-

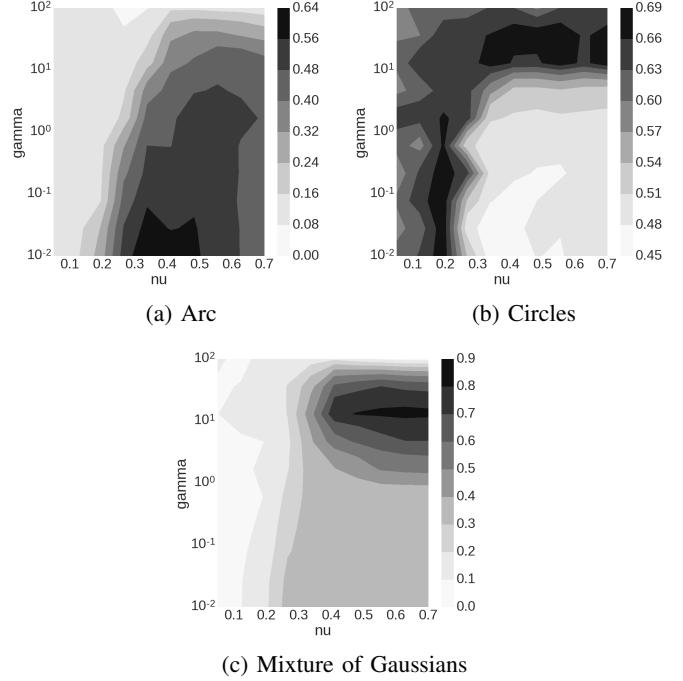


Fig. 2: Synthetic datasets. Proportion of test sample patterns, marked as anomalous by One-Class SVM+

Dataset/Method	One-Class SVM	One-Class SVM+
Arc	0.25	0.67
Circles	0.56	0.96
Mixture of Gaussians	0.55	0.98

TABLE I: Synthetic datasets. Accuracy of anomaly detection

Class SVM. The idea is that if the privileged information does not provide any improvement over the original information, the parameter γ should be set to some big value and the “privileged” part of optimization problem (4) will not have any influence on the overall solution. We provide typical values of the anomaly detection accuracy for the One-Class SVM+ in figure 4. We can see that the privileged information allows obtaining significant increase of the area under the precision/recall curve.

Accuracy of anomaly detection for synthetic datasets is reported in table I.

D. Microsoft Malware Classification Challenge

Zero-day cyber attacks such as worms and spy-ware are becoming increasingly widespread and dangerous. The existing signature-based intrusion detection mechanisms are often not sufficient in detecting these types of attacks. As a result, anomaly intrusion detection methods have been developed to cope with such attacks. Among the variety of common anomaly detection approaches [17], [18], the support vector machine is known to be one of the best machine learning algorithms to classify abnormal behaviours [19]. In this section we demonstrate the applicability of the proposed One-Class

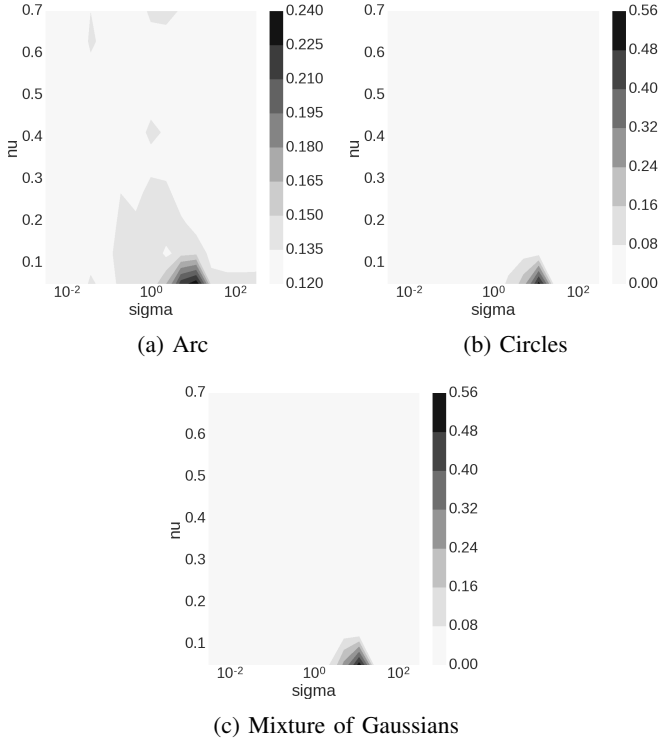


Fig. 3: Synthetic datasets. Area under the precision/recall curve for the original One-Class SVM

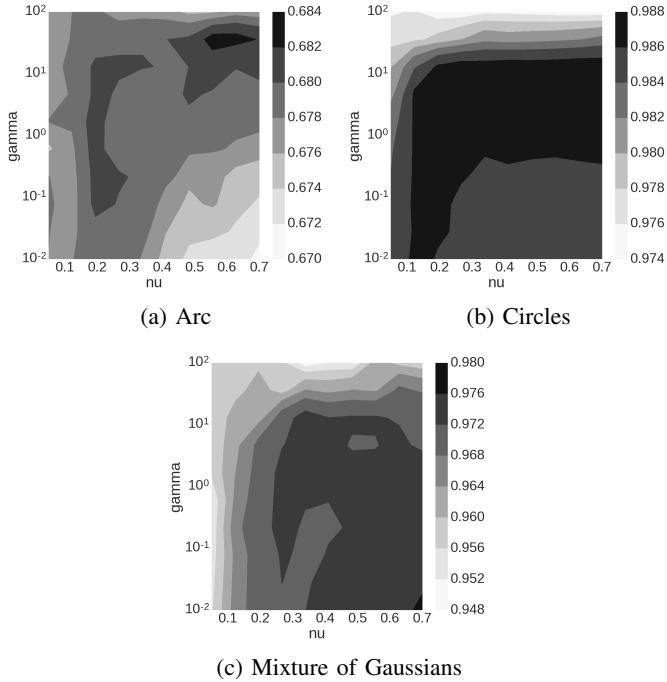


Fig. 4: Synthetic datasets. Area under the precision/recall curve for the One-Class SVM+

SVM modification to cyber attacks detection using a real data from the Microsoft Malware Classification Challenge [14].

In the framework of the Microsoft Malware Classification Challenge a set of known malware files, representing a mix of nine different malware families, is provided. Each malware file has an Id, a 20 character hash value uniquely identifying the file, and a Class, an integer representing one of nine family names to which the malware may belong: Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, Gatak. For each file, the raw data contains the hexadecimal representation of the file's binary content, without the PE header (to ensure sterility). A metadata manifest is also provided, which is a log containing various metadata information extracted from the binary, such as function calls, strings, etc. This was generated using the IDA disassembler tool. The task, proposed to the participants of the challenge, was to develop the best mechanism for classifying files in the test set into their respective family affiliations using binary files and an assembly code.

In order to test our approach to anomaly detection with privileged information we use the same methodology of feature generation, as that initially proposed by the winning team [20]. Using binary files we calculate frequencies of bytes and number of different four-grams. From the assembly code we calculate frequencies of each command, a number of calls to external dll files. Also we use transformation of the assembly code to an image, since malwares can be visualized as grayscale images from byte files or from asm files [21], [22]: each byte is from 0 to 255 so it can be easily translated into pixel intensity. Details of the transformation can be found in [20]. Finally, as the original features we use information, obtained from the binary files, and as the privileged information we use features, obtained from the assembly code. In such a way we want to model a situation, when we have resources to perform reverse-engineering of a program in order to construct the training sample, but we can not make it during the test phase e.g. due to restrictions on computational resources.

For each of the nine malware classes we consider the following problem statement:

- We select one of the nine classes,
- As a train set we use half of patterns from the selected class,
- As a test set we use patterns from another half of the selected class, as well as patterns from other eight classes. We consider patterns from the selected class to be “normal” and patterns from other classes as “abnormal”,
- We use predictions on the test set to calculate the area under the precision/recall curve, representing the accuracy of anomaly detection.

We do not want to claim that such setup of experiments is indeed fully reflects a specificity of cybersecurity applications. In fact through experiments on this real malware data we would like to show that the trained model with privileged information can perform better, and so our approach is useful for cyber security applications.

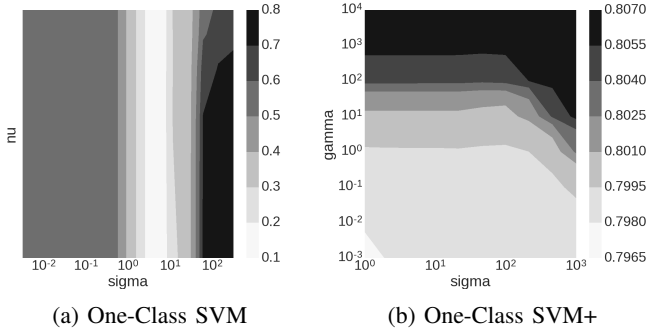


Fig. 5: Malware Detection problem. Dependence of anomaly detection accuracy on the regularization parameters ν and γ , and on the Gaussian kernel widths σ and σ^*

Algorithm/Malware Class	1	2	3	4	5
OneClassSVM	0.67	0.93	0.97	0.69	0.57
OneClassSVM+	0.81	0.95	0.99	0.72	0.57

Algorithm/Malware Class	6	7	8	9
OneClassSVM	0.80	0.82	0.84	0.60
OneClassSVM+	0.81	0.85	0.87	0.62

TABLE II: Malware Detection problem. Comparison of the One-Class SVM and the One-Class SVM+ accuracies

As in the previous subsection, here we also perform comparison with the original One-Class SVM. In figure 5 we provide an example of how the area under the precision/recall curve depends on the parameters. We can notice that it almost does not depend on the regularization parameter ν .

Values of the parameters ν and σ , selected for the One-Class SVM by the cross-validation procedure, are also used for the One-Class SVM+. Thus, for the One-Class SVM+ we only tune values of the regularization γ and the kernel width σ^* in the privileged feature space. We expect that in case of a small kernel width σ^* and a big value of γ we obtain results similar to that of the One-Class SVM.

We provide obtained results in table II. For some malware classes the privileged information allows getting significant increase in accuracy of anomaly detection. For other malware classes accuracies of One-Class SVM and One-Class SVM+ turned out to be the same thanks to the fact, that for specific values of the kernel width in the privileged feature space and specific values of the regularization parameter the decision function of the One-Class SVM+ is close to the decision function of the original One-Class SVM.

E. Comparison with Related Approaches

In [15] and [16] (we provide the review of the related methods in section IV) authors provided results of experiments on real datasets. In particular, in [15] results of experiments on four datasets are described, and in [16] only two datasets are used among those, which are considered in [15]. In order to compare approaches from [15] and [16] with methods, proposed in this paper, we use one of these two samples —

Feature/Method	One-Class SVM+ [15]	SVDD+ [16]	One-Class SVM+
Length	0.670	0.673	0.702
Height	0.730	0.728	0.751
Whole weight	0.715	0.710	0.744

TABLE III: Comparison with related approaches. Accuracy Values

the sample Abalone. Results of experiments for other datasets are similar.

The Abalone dataset contains size and weight of molluscs, as well as their age, evaluated from a number of rings on a shell. Patterns are divided into two groups depending on the value of the parameter “Rings”. Patterns with Rings < 7 are considered as a normal class, the rest patterns are considered to be anomalies.

Let us construct several blocks of a privileged information. For this we divide the dataset into two groups w.r.t. to parameters “Length” (Length < 0.5), “Height” (Height < 0.15) and “Whole weight” (Whole weight < 0.8). Each such division can be encoded by a binary vector, which we use as an additional information during the training phase. We perform three experiments. In each of the experiments we use one of these blocks of privileged information. We evaluate classification accuracy by the ten-fold cross-validation procedure.

In the previous sections we used the area under the precision/recall curve in order to evaluate the performance of anomaly detection algorithms. Unfortunately, in [16] the authors provided only values of the accuracy measure, therefore in this section for comparability we also provide only values of the accuracy measure. Results are given in table III. We can see that the parametrization of the One-Class SVM+, proposed in this paper, allowed us to find more efficient solution. Also let us note that results of the One-Class SVM+ from [15] and of the SVDD+ from [16] are comparable.

VI. CONCLUSIONS

We provide modifications of the approaches for one-class classification problem that allows to incorporate a privileged information. We can see from the results of experiments that in some cases the privileged information can significantly improve the anomaly detection accuracy. In cases when the privileged information is not useful for the problem at hand thanks to the structure of the corresponding optimization problem (e.g. cf. (1) with (4)) the privileged information will not have a significant influence on the decision function: e.g. since for $\gamma \gg 1$ the solution of (4) is close to the solution of (1), then the decision function of the One-Class SVM+ is close to the decision function of the original One-Class SVM.

VII. ACKNOWLEDGEMENTS

The research of the first author was supported by the RFBR grants 16-01-00576 A and 16-29-09649 ofi_m. The research of the second author was conducted in IITP RAS and supported solely by the Russian Science Foundation grant (project 14-50-00150).

REFERENCES

- [1] Manevitz, L.M., Yousef, M.: One-class svms for document classification. *Journal of Machine Learning Research* **2** (2001) 139–154
- [2] Khan, S.S., Madden, M.G.: One-class classification: Taxonomy of study and review of techniques. *The Knowledge Engineering Review* **29**(03) (2014) 345–374
- [3] Khan, S.S., Madden, M.G.: A survey of recent trends in one class classification. In: *Proceedings of the 20th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, in the LNAI. Volume 6206. Springer-Verlag* (2009) 181–190
- [4] Tax, D.M.J., Duin, R.P.W.: Support vector data description. *Mach. Learn.* **54**(1) (January 2004) 45–66
- [5] Chang, W.C., Lee, C.P., Lin, C.J.: A revisit to support vector data description. Technical report (2013)
- [6] Schölkopf, B., Williamson, R.C., Smola, A.J., Shawe-Taylor, J., Platt, J.C.: Support vector method for novelty detection. In Solla, S.A., Leen, T.K., Müller, K., eds.: *Advances in Neural Information Processing Systems 12. MIT Press* (2000) 582–588
- [7] Burnaev, E., Erofeev, P., Smolyakov, D.: Model selection for anomaly detection. In Verikas, A., Radeva, P., Nikolaev, D., eds.: *Proc. SPIE 9875. Volume 9875. SPIE* (2015)
- [8] Burnaev, E., Erofeev, P., Papanov, A.: Influence of resampling on accuracy of imbalanced classification. In Verikas, A., Radeva, P., Nikolaev, D., eds.: *Proc. SPIE 9875. Volume 9875. SPIE* (2015)
- [9] Artemov, A., Burnaev, E.: Ensembles of detectors for online detection of transient changes. In Verikas, A., Radeva, P., Nikolaev, D., eds.: *Proc. SPIE 9875. Volume 98751Z. SPIE* (2015)
- [10] Kuleshov, A., Bernstein, A.: Incremental construction of low-dimensional data representations. In: *Lecture Notes in Artificial Intelligence, “Artificial Neural Networks for Pattern Recognition”*. Volume 9896. Springer Heidelberg (2016) 13 pp.
- [11] Bernstein, A., Kuleshov, A., Yanovich, Y.: Information preserving and locally isometric and conformal embedding via tangent manifold learning. In: *Proceedings of the International IEEE Conference on Data Science and Advanced Analytics (DSAA 2015). IEEE Computer Society, Piscataway, USA* (2015) 1–9
- [12] Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. *Neural Comput.* **12**(5) (May 2000) 1207–1245
- [13] Pechyony, D., Vapnik, V.: On the theory of learning with privileged information. In Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A., eds.: *Advances in Neural Information Processing Systems 23. Curran Associates, Inc.* (2010) 1894–1902
- [14] : Microsoft malware classification challenge (big 2015). <https://www.kaggle.com/c/malware-classification>
- [15] Zhu, W., Zhong, P.: A new one-class svm based on hidden information. *Knowledge-Based Systems* **60** (2014) 35–43
- [16] Zhang, W.: Support vector data description using privileged information. *Electronics Letters* **51**(14) (2015) 1075–1076
- [17] Nissim, N., Moskovitch, R., Rokach, L., Elovici, Y.: Detecting unknown computer worm activity via support vector machines and active learning. *Formal Pattern Analysis and Applications* **15**(4) (2012) 459–475
- [18] Nissim, N., Moskovitch, R., Rokach, L., Elovici, Y.: Novel active learning methods for enhanced pc malware detection in windows os. *Expert Systems with Applications* **41**(13) (2014) 5843–5857
- [19] Shon, T., Moon, J.: A hybrid machine learning approach to network anomaly detection. *Information Sciences* **177** (2007) 3799–3821
- [20] Wang, X., Liu, J., Chen, X.: Microsoft malware classification challenge (big 2015). first place team: Say no to overfitting. https://github.com/xiaozhouwang/kaggle_Microsoft_Malware/blob/master/Saynootoverfitting.pdf
- [21] Nataraj, L. <http://sarvamblog.blogspot.com/> (2014)
- [22] Nataraj, L., Karthikeyan, S., Jacob, G., Manjunath, B.: Malware images: visualization and automatic classification. In: *Proceedings of the 8th International Symposium and Visualization for Cyber Security. ACM* (2011) 7 pp.